

Cerebrate

A quick glance at the Cerebrate architecture

Andras Iklody

Internal



- Review the following:
 - ▶ The layout of the codebase
 - ▶ Some design principles
 - ▶ Architecture of the application
- Generally anything that might be relevant when reviewing the codebase

- CakePHP 4.x
- Bootstrap 4 UI
- A host of ported and modernesied MISP libraries
- Database: MySQL / raw file storage
- Redis will be added in the future

- MVC design
- API / UI dual design
- ReSTful APIs
- Heavy on abstraction

Cerebrate ContactDB ▾ Trust Circles ▾ Administration ▾ Cerebrate ▾ Open ▾ Logout

Individuals

- List individuals
- Add individual



Organisations

- List organisations
- Add organisation
- View organisation**
- Edit organisation
- Delete organisation

Encryption keys

- List encryption keys

Organisation View

ID	2
Name	CIRCL
UUID	55f6ea5e-2c60-40e5-964f-47a8950d210f
URL	
Nationality	Luxembourg
Sector	Security
Type	
Contacts	
Alignments	[Code monkey] andras.iklody@gmail.com  [Head honcho] alexandre.dulaunoy@circl.lu 

Add individual

4

17

BASIC DESIGN PRINCIPLES - API

← → ↻ ⓘ localhost:8000/organisations/view/2.json

```
{
  "id": 2,
  "uuid": "55f6ea5e-2c60-40e5-964f-47a8950d210f",
  "name": "CIRCL",
  "url": null,
  "nationality": "Luxembourg",
  "sector": "Security",
  "type": "",
  "contacts": null,
  "alignments": [
    {
      "id": 4,
      "individual_id": 1,
      "organisation_id": 2,
      "type": "Code monkey",
      "individual": {
        "id": 1,
        "uuid": "b571bc22-b9af-44c7-b4b2-1d57e669efa6",
        "email": "andras.iklody@gmail.com",
        "first_name": "Andras",
        "last_name": "Iklody",
        "position": "Code monkey"
      }
    }
  ],
  {
    "id": 13,
    "individual_id": 5,
    "organisation_id": 2,
    "type": "Head honcho",
    "individual": {
      "id": 5,
      "uuid": "005896e5-dd4b-4822-8ef0-091b236e91c2",
      "email": "alexandre.dulaunoy@circl.lu",
      "first_name": "Alexandre",
      "last_name": "Dulaunoy",
      "position": "Head honcho"
    }
  }
}
```

DIRECTORY STRUCTURE

- src
 - ▶ Command
 - ▶ Controller
 - ▶ Model
 - ▶ View
- templates
- libraries

- Inheritance of the ApplicationController
- Reusable code via components
 - ▶ CRUDComponent
 - ▶ ACLComponent
 - ▶ ParamHandlerComponent
 - ▶ RestResponse Component

- Generally access public functions by /controller/action
- Response is always split between two paths: UI/API
- We can therefore test all functionalities via the API, the UI uses the same program logic

- Tables
- Entities
- Tables Implement AppTable
- Entities implement AppModel
- Reusable code: Behaviors

- Reusable code in Views
- Templates
 - ▶ UI factories for change management and uniformity
 - ▶ Parametrised view templating
- Exception: API does not use views, rather serializes response data in the controller

- The plan is to feature a wide range of tools for the CLI
- Currently it's only in use for one use-case (password resets)
- CLI commands directly interact with the model, no controller code is executed

CLI TOOLS

```
What would you like to do?
> 1
-----+-----
| ID | Username | Email |
-----+-----+
| 4 | Iglocska | andras.iklody@gmail.com |
| 5 | user     | andras.iklody@gmail.com |
-----+-----+
| Cerebrate users |
-----+-----+
| 1 | List users |
| 2 | Reset password for a user |
| 0 | Exit |
-----+-----+
What would you like to do?
> 2
Which user do you want to reset?
> 4
Would you like to generate a password automatically for user "Iglocska"? (y/n)
> n
Please enter the desired password:
> Password1234
Password reset for user "Iglocska". The new password is: "Password1234"
-----+-----+
| Cerebrate users |
-----+-----+
| 1 | List users |
| 2 | Reset password for a user |
| 0 | Exit |
-----+-----+
What would you like to do?
> █
```

■ Authentication

- ▶ Username / API key
- ▶ Will be extended in the future, pending also requirement collection

■ RBAC

- ▶ Users are tied to role objects
- ▶ Roles are modifiable permission control tables
- ▶ The ACL component uses the user session's role to determine access for the requested endpoint
- ▶ We deny by default

AUTHENTICATION AND RBAC

The screenshot displays the Cerebrate administration interface. At the top, a navigation bar includes the Cerebrate logo and several dropdown menus: ContactDB, Trust Circles, Administration, Cerebrate, Open, and Log out. On the left, a sidebar menu under the heading 'Roles' contains the following options: List roles, Add role, View role (which is highlighted in blue), Edit role, and Delete role. The main content area is titled 'Role view' and displays the details for a specific role:

ID	4
Name	Admin
Admin permission	✓
Default role	✓

- Currently in early stages
- Opt-in system to publicly disclose organisation lists (act as a public trust authority)
- Unauthenticated endpoints, controlled by the configuration file

- Potential point to test: JSON libraries ingested by Cerebrate
- Currently one use-case - extending models with meta fields

PLANS FOR THE FUTURE IN TERMS OF MAJOR SUBSYSTEMS

- Cerebrate to Cerebrate exchange, similar to the MISP synchronisation
- Cerebrate to local tool exchange using a modular approach
- Trust relationships with a direct ingestion and inbox system tier